REMARKS

Pending claims

Assuming entry of this amendment, claims 2-4, 6-9, 12-17, and 20-33 are still pending, of which claims 12, 13, 21, 22, 27 and 30 are independent.

Specification Amendment

The change in the specification is to correct an obvious typographical error and adds no new matter.

Voluntary Claim Amendment

Claim 8 has been amended to correct an obvious misspelling.

Claim Rejections – Rejections Under 35 U.S.C. 102

The Examiner rejected all of the pending claims under 35 U.S.C. 102(e) as being anticipated by US 6,725,456 ("*Bruno*"). Tracking claim 21, the Examiner wrote:

> Bruno teaches a computer system comprising:
>
> a host system which includes a host operating system and a hardware memory that is addressable in a hardware memory address space (see col. 31ines 15-55);
>
> at least one virtual computer, each computer includes at least one virtual processor, guest physical memory, and a guest OS operable to address, allocate and de-allocate the guest physical memory in a guest physical address space and is operatively connected to the host system (see col. 31ines 15-55 and col. 61ines 55-col. 7 lines 5);
>
> a memory reservation software module located within the virtual computer for receiving a memory quantity request from the host system and for changing the allocation of the guest physical memory from within the respective guest OS according to the memory quantity request thereby changing the amount of the hardware memory available for arbitrary use by the host system (see col. 3 lines 15-55).

As a general matter, *Bruno* is focused on proportional-share scheduler ***policies***, and on exporting a uniform API for them. In contrast, the applicant's invention relates to a low-level ***mechanism***, which can be used to support higher-level policies; the policy need not be proportional sharing or reservation-based, as in *Bruno*.

The Examiner's analysis is incorrect on several specific points as well:

First, the invention as defined in all of the independent claims 12, 13, 21 and 27 include a limitation with wording at least substantially identical to that used in claim 21: "a memory reservation software module ***located within the virtual computer*** for ***receiving*** a memory quantity request ***from the host system*** and for changing the allocation of the guest physical memory ***from within the respective guest OS*** according to the memory quantity request, thereby changing the amount of the hardware memory available for arbitrary use by the host system." Independent claims 22 and 30 include an analogous limitation, but in which a "guest system" is recited instead of a virtual computer.

In claim 22, the resource is a number of processors instead of memory. Moreover, although independent claims 12, 13, 21 and 27 recite a guest operating system, the invention as defined in claims 22 and 30 is not so limited. (As stated in claim 32, the memory reservation software module may also be a user-level application loaded in the guest system.) For the sake of simplicity, however, consider the formulation found in claim 21; the following comments apply equally to the other independent claims.

According to the invention, the request for a change in the quantity of memory made available for use by the virtual computer is passed from the underlying host to the virtual computer. A mechanism located within the virtual computer itself then takes actions that cause a change in the virtual computer's resource allocation. In *Bruno*, requests for resources are passed in the opposite direction, from the applications to schedulers located within the Eclipse/BSD operating system; the schedulers then handle the requests. See, for example, *Bruno's* Figure 1, as well as:

Col. 5, lines 1-4 (emphasis added):

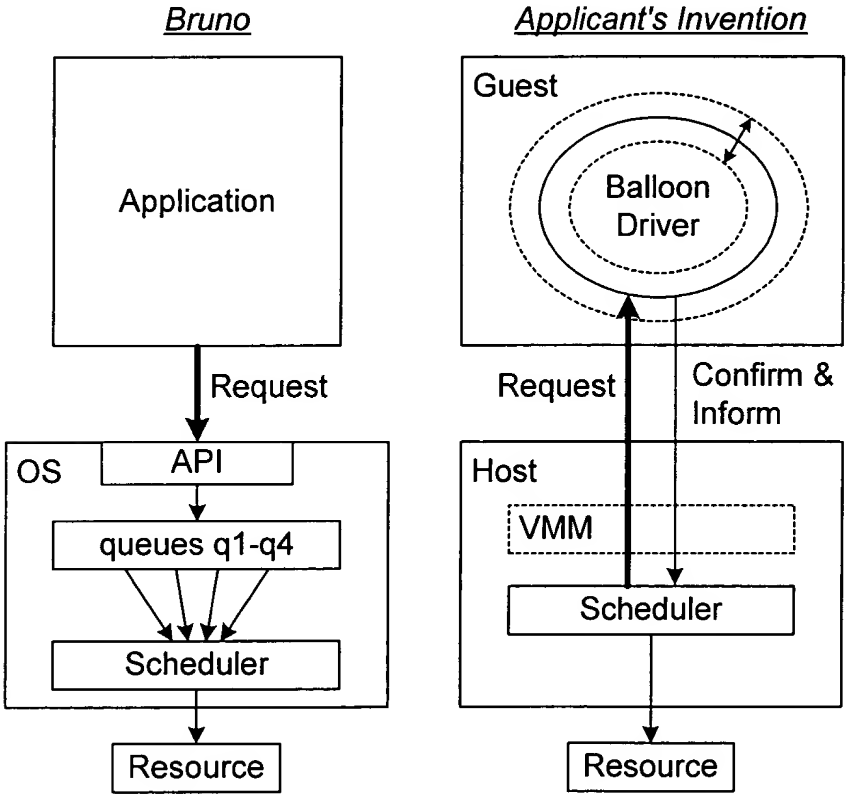
> ... every request ***arriving at*** a given one of the above-noted ***schedulers*** must specify a queue, and the given ***scheduler*** apportions resources to each queue based on the queue's share of that resource.

and

Col. 4, lines 38-39:

> The invention provides techniques for integrating proportional share schedulers ***into conventional operating systems***

The following simplified figure illustrates this difference between *Bruno* and the applicant's invention as claimed:



Thus, in *Bruno*, resource requests are initiated by and passed from the software entity (the application) running on the underlying system-level components (the operating system), ***down*** (as illustrated here) to the system-level schedulers. In contrast, in the applicant's invention, it is the system-level component that initiates a request for a change in resource allocation by signaling ***up*** to a component (the "balloon" driver or application) within the guest (virtual computer or other), which in turn invokes whatever mechanism is provided by the guest itself for changing the allocation of the resource of interest. Although the guest responds, this is to communicate whether the received request was completed successfully, not to pass on the initial request itself. This difference has significant consequences.

As just one example of the consequences, although different in some aspects from a classic "application-level paging" system, *Bruno* suffers from many of the same weaknesses described in connection with this form of prior art, as mentioned in the specification on page 7, lines 5-8:

> One disadvantage of application-level paging is that it requires an explicit interface and protocol to allow the OS and the various applications to cooperate in selecting pages to be relinquished. The page revocation mechanism in application-level paging is therefore not transparent to the OS.

Similarly, to facilitate communication between applications and the operating system's schedulers, *Bruno* requires all applications using his system to address (and therefore be specifically designed or modified to address) a specialized application program interface (API). See, for example,

Col. 3, lines 52-55:

> the invention minimizes the number of modifications that may be necessary in existing applications for them to be able to benefit from proportional-share scheduling.

and

Col. 2, lines 33-43:

> First, it is desirable that an operating system provide a uniform application programming interface (API) for all of the system's schedulers and resources. In the case of proportional share schedulers, this should be a resource reservation API, which allows applications to reserve for exclusive use portions of each resource. However, several of the above-mentioned proportional share schedulers were proposed without an API ...
>
> Second, it is desirable that the resource reservation API be easy to integrate with the conventional API of existing operating systems ...

Still another weakness of systems such as *Bruno's* is described on page 5, lines 15-19, of the applicant's specification:

> One disadvantage of existing dynamic resource allocation mechanisms is that the guest operating system running, for example, in each VM has much better information about which of its allocated memory pages (or analogous units of memory) contain the least time-critical data and are therefore more suitable to be swapped out to a slower storage device such as the hard disk

Thus, if the availability of a resource (such as memory pages) to an application is to be *reduced* in *Bruno*, then the decision as to which pages are to be revoked will be left up to the scheduler 33. Where the "application" is a virtual computer or other guest with its own operating system, the applicant's invention allows this guest operating system to make its own decision, using its own rules and local knowledge, as to which pages to swap out, with no need to modify the kernel of the guest operating system (or other allocation software) at all. Although the mechanism for making this possible is described in detail in many places in the specification, additional advantages of this arrangement are summarized in the "Advantages of the Invention" section at the end of the specification.

Assume that the underlying host system needs more of the resource, for example, more memory. Using the invention, the host can tell the balloon driver (or application) in the guest to *inflate*, which will make the guest operating system believe that the driver needs more memory for itself. Using whatever rules it has been designed to implement (as opposed to rules in the host that are imposed on it), the guest then swaps out memory pages to disk (including a *virtual* disk in a virtualized system), if possible or desirable, thereby freeing these memory pages for use by the balloon driver.

The driver does not actually need these pages for itself, however, but rather is simply causing them to be removed from the set of pages used within the guest. These pages can instead be used by the *host* for whatever purpose that led it to request more pages in the first place. Thus, the guest will have relinquished pages to the host, but without realizing this. As the independent claims state, the change in allocation of the resource is done from within the respective guest according to the memory quantity request, "thereby changing the amount of the [resource] available for arbitrary use by the host system."

In contrast, *Bruno* does not allow the application any control over which of its pages (or units of any other resource) are to be revoked. In fact, *Bruno* does not address the issue of resource revocation at all. Rather, *Bruno* includes a mechanism only for "destroying resource reservations" and then only when a process exits (col. 7, lines 51-59). The invention allows the guest to dynamically revoke the resource (such as memory pages) even for running processes.

Another big difference between *Bruno* and the applicant's invention as claimed in independent claims 12, 13, 21 and 27 involves just what software entity is running on the underlying system-level software. *Bruno* involves "an **application** running on an operating system" (col. 3, lines 16-17, emphasis added). As is well known, the code defining a virtual computer is ultimately executed on some hardware platform, with at least partial mediation by some system-level software component such as an operating system. As such, viewed from the perspective of the underlying hardware and operating system a virtual computer may appear as an application, but the independent claims 12, 13, 21 and 27 make clear additional features of a virtual computer that are not mentioned in *Bruno* and, indeed, would appear to be irrelevant to *Bruno's* system. These features include "at least one virtual processor, guest physical memory, and a guest OS operable to address, allocate and de-allocate the guest physical memory in a guest physical address space ..." No such software components are mentioned in *Bruno's* "application."

Because the independent claims all recite features of the invention that are not found at all in *Bruno*, and that provide clear and well-described advantages, the applicant respectfully submits that these claims, as well as all the claims that depend from them, should be allowable without any substantive amendment relative to the previous amendment. Nonetheless, the applicant wishes to point out examples (not exhaustive) of features recited in the dependent claims that are lacking in *Bruno*.

For example, the Examiner wrote (emphasis added):

> "As to claim 2, Bruno teaches [that] the memory reservation software module is a ***driver*** installed within each respective guest operating system (see col. 31lines 15-55)"
>
> "[a]s to claim 3, "Bruno teaches … a communications means for communicating a respective memory quantity request to each ***driver***"
>
> "[a]s to claim 4, Bruno teaches … ***guest operating system*** memory reservation means for reserving specified amounts of the system memory (see col. 31lines 15-55)" and that "the ***driver*** is operatively connected to the memory reservation means …"

The applicant respectfully disagrees: *Bruno* does not mention any operating system other than the one that in interfaces with the actual hardware. Thus, *Bruno* has a "host" operating system in the sense of this invention, but no "guest" operating system at all.

As to claim 6, the Examiner wrote:

> "Bruno teaches [a] … virtual machine monitor forming an interface between the memory scheduler and each respective virtual machine."

Nowhere in *Bruno* is the concept of a virtual machine or virtual machine monitor mentioned at all.

Request to Withdraw Finality of Office Action

The Examiner made this second Office action final and rejected all of the pending claims in view of the *Bruno*. MPEP 706.07(a) (Final Rejection, When Proper on Second Action) states, in pertinent part:

> Under present practice, second or any subsequent actions on the merits shall be final, except where the examiner introduces a new ground of rejection that is neither necessitated by applicant's amendment of the claims nor based on information submitted in an information disclosure statement … filed during the period set forth …

The applicant respectfully submits that the reason for the new ground of rejection (the *Bruno* reference) is not because of any claim amendment on the part of the applicant, but rather because *Bruno* – which issued on 20 April 2004 – was not even available as a published reference when the Examiner issued the first Office action, or when the applicant responded to that Office action. Not to give the applicant a chance to explain the patentability of his invention over *Bruno,* without filing a request for continued examination and paying another filing fee, would be unfair: The applicant cannot reasonably be expected to respond to prior art that he cannot possibly have known about.

As explained above, the claims in this application are patentable over *Bruno* even as previously presented (with one small error in typing in claim 8). Moreover, no new issues are raised by the claims. If the Examiner does not allow the current claims, then the applicant requests another substantive Office action on the merits of the claims rather than a routine Advisory Action.

Conclusion

The independent claims recite advantageous features of the invention that are not found at all in the newly cited reference. As such, the independent claims should be allowable over the cited prior art. The various dependent claims of course simply add additional limitations and should therefore be allowable along with their respective independent base claims.

Change of Attorney Name

Along with the previously submitted amendment was a letter giving notice of a change of the attorney's last name from "Slusher" to "Pearce." From the address of this 19 July Office action it appears that this has not yet been reflected in the Office's record for this application. (The name Pearce has been changed for registration number 34,729 for more than a year.) Please update the record for this application. Thank you.

Date: 9 August 2004

Respectfully submitted,

Jeffrey Pearce

Jeffrey Pearce
Reg. No. 34,729
Attorney for the Applicant

34825 Sultan-Startup Rd.
Sultan, WA 98294
Phone & fax: (360) 793-6687